CP: Generating an EXE file

Presented by: K.Aryan
B23IT117

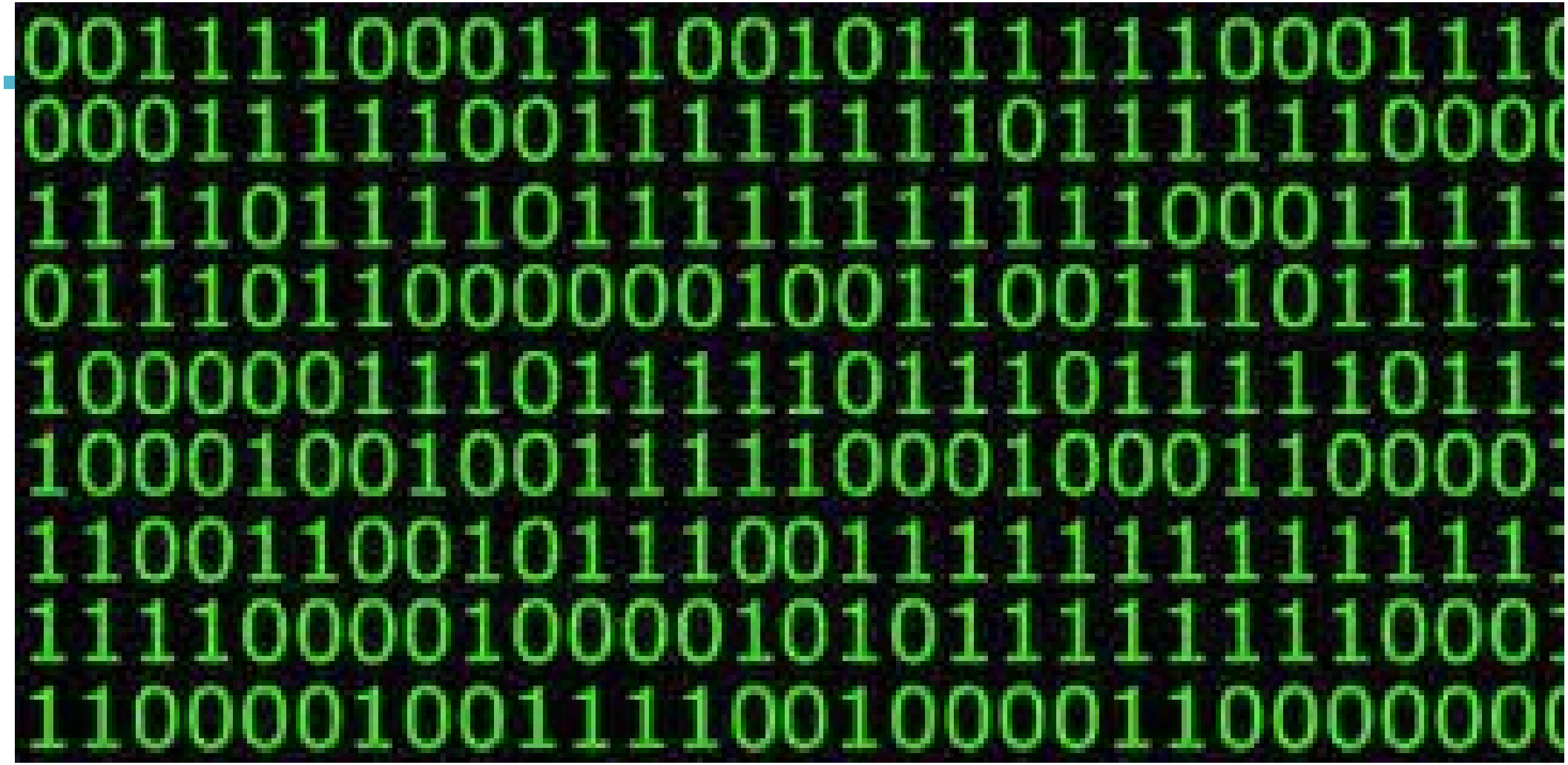
```c
#include <stdio.h>
int main() {
    int number1, number2, sum;

    printf("Enter First Number: ");
    scanf("%d", &number1);

    printf("Enter Second Number: ");
    scanf("%d", &number2);

    // calculating sum
    sum = number1 + number2;

    printf("\nAddition of %d and %d is %d", number1, number2, sum);
    return 0;
}
```


0011110001110010111111000111(
0001111100111111101111110000(
1111011110111111111111000111
0111011000000100110011101111
1000011101111011101111101111
1000100100111110001000110000
1100110010111001111111111111
1111000010000101011111110001
1100001001111001000011000000

EXE

Method of generating exe files using list variables

- Powerful and flexible
- Simplifies development

How its done

1. Subdividing the target name
2. Saving atleast one portion as list variable
3. Determining if the target can be further subdivided
4. Using list variable as a run set component

Subdividing target name

- Target name name of exe file
- Subdividing the target name at predetermined points
- predetermining points underscores or periods

Saving atleast one portion as a list variable

- Saving one portion as a list variable
- list variable data structure that can store a list of items

Determining if the target name can be further subdivided

- Identifying delimiters and using them to subdivide further
- Eg: my_program.exe can be divided into “my_program” and “exe”

Using list variable as a run set components, manufacturing set components, or compiler condition

- We can use the list variable as a runset component, manufacturing set components or compiler condition for generating exe

- Run set components tasks needed to be performed to generate exe

- Manufacturing process components steps needed to generate exe

- compiler condition controls the compiler regarding exe generation